

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-338959

(43)Date of publication of application : 08.12.2000

(51)Int.Cl. G09G 5/36  
G06T 1/20  
G06T 11/20

(21)Application number : 11-152702

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 31.05.1999

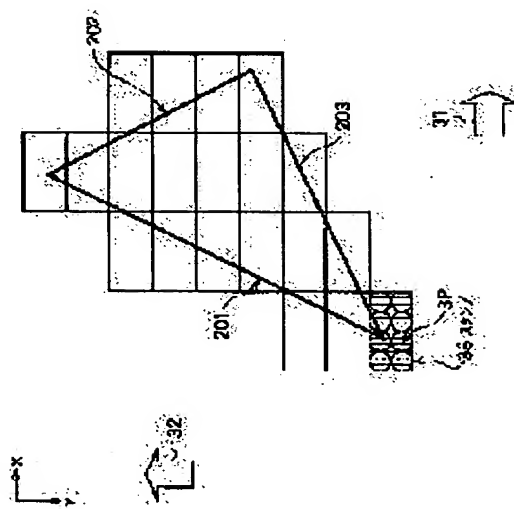
(72)Inventor : KOIZUMI TOMOHIRO  
TAKEMOTO TAKU  
TAKENAKA YASU HARU

## (54) IMAGE PROCESSING DEVICE

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To reduce hardware resources by performing a scan along lines parallel to the X-axis and Y-axis of a screen coordinate system in the area unit consisting of at least one pixel to generate the pixel in the area unit within an area subject to drawing.

**SOLUTION:** An object to be drawn is a triangle formed of a starting side 201 and ending sides 202 and 203. The starting side means the side where a scan along X-axis comes in the object, and the ending sides mean two ends where the scan comes out. Each rectangular area shows the stamp of pixels to be simultaneously processed, and a pixel 3P is the pixel of a DDA drawing start point coordinate. The scan is advanced from a scan DDA starting stamp 33 in the positive direction 31 of X to end the scan of this line in a certain point, and transferred to the next line (advanced in the negative direction 32 of Y). This is repeated to scan all the area of the object.



## LEGAL STATUS

[Date of request for examination]	27.01.2003
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	abandonment
[Date of final disposal for application]	31.03.2006
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-338959

(P2000-338959A)

(43) 公開日 平成12年12月8日 (2000. 12. 8)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テマコード* (参考)
G 0 9 G 5/36	5 1 0	G 0 9 G 5/36	5 1 0 V 5 B 0 5 7
G 0 6 T 1/20		G 0 6 F 15/66	K 5 B 0 8 0
11/20		15/72	3 5 5 K 5 C 0 8 2

審査請求 未請求 請求項の数 9 O L (全 15 頁)

(21) 出願番号 特願平11-152702

(22) 出願日 平成11年5月31日 (1999. 5. 31)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 小泉 友弘

神奈川県川崎市幸区堀川町580番1号 株式会社東芝半導体システム技術センター内

(72) 発明者 竹本 卓

神奈川県川崎市幸区堀川町580番1号 株式会社東芝半導体システム技術センター内

(74) 代理人 100083806

弁理士 三好 秀和 (外7名)

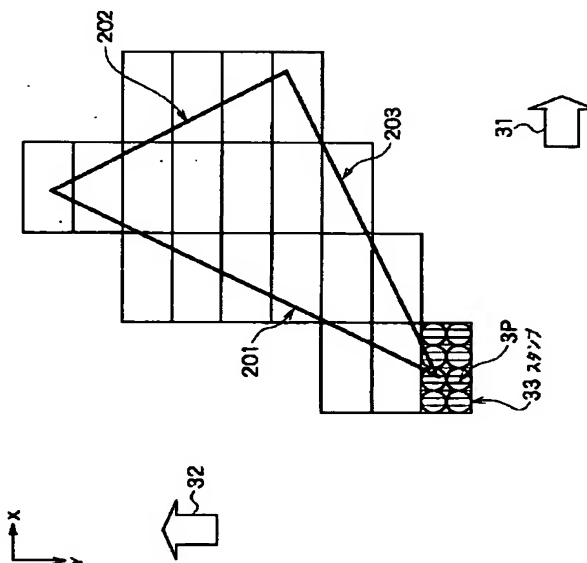
最終頁に続く

(54) 【発明の名称】 画像処理装置

(57) 【要約】

【課題】 ピクセルを発生させる線形補間処理を行うためのハードウェアリソースを低減すること。

【解決手段】 描画オブジェクト内に、ピクセルを矩形領域単位で発生し、しかも、前記矩形領域単位でピクセルを発生する際のスキヤンの仕方は前記オブジェクトのX方向のスキヤンラインに沿って、スキヤン開始点からピクセルを発生し、前記描画オブジェクトの境界に達すると、Y方向にスキヤンラインを一段移動して、スキヤン開始点側からスキヤンラインに沿ってピクセルを発生して行く。即ち、前記描画オブジェクト内をX方向、Y方向にジグザグにスキヤンしてピクセルを矩形領域単位で発生する。これにより、ピクセルを発生させるための線形補間の計算が加減算のみで出来るため、線形補間処理のハードウェアを加減算器のみで構成でき、従来の乗算器が必要であった場合に比べて、ハードウェアリソースを低減させることができる。



## 【特許請求の範囲】

【請求項 1】 スクリーン座標系の描画対象領域内にピクセルを発生する画像処理装置において、少なくとも 1 個以上のピクセルから成る領域単位で、前記スクリーン座標系の X 軸、Y 軸に平行なラインに沿ってスキャンすることにより、前記描画対象領域内に前記領域単位でピクセルを発生する手段を具備することを特徴とする画像処理装置。

【請求項 2】 前記スキャンに伴い、次に発生するピクセルを含む領域単位が前記描画対象領域内に入るか否かの領域判定を行う領域判定手段を具備し、少なくとも 1 個以上のピクセルが前記描画対象領域内に入る場合に、前記領域単位でピクセルを発生することを特徴とする請求項 1 記載の画像処理装置。

【請求項 3】 前記領域判定手段は、直近で発生したピクセルを含む領域の外に設定する先行点の前記描画対象領域内に入るか否かで前記領域判定を行うことを特徴とする請求項 2 記載の画像処理装置。

【請求項 4】 前記領域内の少なくとも 1 個以上の特定のピクセルのデータと前記領域内の残りのピクセルのデータとを算出する際の計算精度を変化させる算出手段を具備することを特徴とする請求項 1 乃至 3 いずれかに記載の画像処理装置。

【請求項 5】 前記領域内のピクセルデータを同時に処理する並列処理手段を具備することを特徴とする請求項 1 乃至 4 いずれかに記載の画像処理装置。

【請求項 6】 前記並列領域を形成する各領域内の少なくとも 1 個以上の特定のピクセルのデータについては精度の高い計算により算出し、前記各領域内の残りのピクセルのデータについては精度の低い計算により算出することを特徴とする請求項 5 記載の画像処理装置。

【請求項 7】 前記領域単位で発生したピクセル各々が前記描画対象領域内に入っているかどうかを判定するマスク情報を発生するマスク発生手段と、前記発生したマスク情報により前記描画対象領域内に入っているピクセルだけを描画する描画手段を具備することを特徴とする請求項 1 乃至 6 いずれかに記載の画像処理装置。

【請求項 8】 前記領域単位を任意の数、複数個連結して形成する並列領域を設定する設定手段と、前記描画対象領域内に前記設定された並列領域単位でピクセルデータを同時に発生する並列処理手段とを具備することを特徴とする請求項 1 乃至 7 いずれかに記載の画像処理装置。

【請求項 9】 前記並列処理手段は、前記領域単位の少なくとも 1 個以上の特定のピクセルを代表点とし、且つこの代表点の位置を前記領域単位の連結個数によらずに一定とし、この代表点のデータと残りピクセルのデータを算出する際の計算精度を変化させることを特徴とする請求項 8 記載の画像処理装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、3 次元画像処理用途としての画像処理装置に係り、特にポリゴン内部のピクセルを発生するピクセル発生装置に関する。

## 【0002】

【従来の技術】 3 次元画像処理装置は 3 次元の座標データをもったオブジェクトを 2 次元空間に投影し、ディスプレイに表示するための仕組みを持っている。3 次元オブジェクトのモデリングを行う上で、ポリゴンとよばれる多角形を用いることがある。このポリゴンが 2 次元空間へ投影されると、ポリゴンの頂点情報からポリゴン内部のピクセル発生処理が行われるが、これはよく DDA (Digital Differential Analyzer) と呼ばれる線形補間手法を用いて実行される。

【0003】 DDA プロセスでは頂点情報からポリゴンの辺方向へのデータの傾きを求め、この傾きを用いて辺上のデータを算出した後、続いてラスタ走査方向 (X 方向) の傾きを算出し、この傾きを用いて内部のピクセルを発生していく。

【0004】 ここで、辺上のある点の座標を (x0, y0) とする。但し、y0 は整数値である。この点の色情報等のデータのひとつを P0 とする。同じラスタライン (X 軸方向) が交差するもう一方の辺上の座標及びパラメータ値を (x1, y0)、P1 とする。このラスタライン上の任意の座標のパラメータ値 P は線形補間手法により、 $P = P0 + dPx * (x - x0)$  となる。

【0005】 ここで、 $dPx = (P1 - P0) / (x1 - x0)$  である。

## 【0006】

【発明が解決しようとする課題】 従来の画像処理装置により上記した式に基づいてピクセルを発生させると、x0 が小数付き座標の場合があるため、どうしても多くのハードウェアリソースを要する乗算器を使う必要がある。しかも、ピクセルを発生させる際、R、G、B 及びテクスチャマッピングで使用する複数のパラメータを同時に算出しなければならず、これらパラメータの数だけ、乗算器が必要となり、ハードウェアの規模がかなり大きくなってしまいう問題があった。

【0007】 また、画像システムによっては高速描画を実現するために並列処理によりピクセルを発生させる手法が用いられることがある。この場合、その並列度に応じて DDA を実行する演算器を増やす必要があり、それに比例して、ハードウェアリソースを増やさなければならず、更にハードウェアが大規模化し、装置のコストを上昇させると共に、装置を大型化してしまうという問題もあった。

【0008】 本発明は、上述の如き従来の課題を解決するためになされたもので、その目的は、ハードウェアリソースを低減することができ、又、高速描画を実現する

ために並列処理する際のハードウェアリソースの増加を緩和することができる画像処理装置を提供することである。

#### 【0009】

【課題を解決するための手段】上記目的を達成するために、請求項1の発明の特徴は、スクリーン座標系の描画対象領域内にピクセルを発生する画像処理装置において、少なくとも1個以上のピクセルから成る領域単位で、前記スクリーン座標系のX軸、Y軸に平行なラインに沿ってスキャンすることにより、前記描画対象領域内に前記領域単位でピクセルを発生する手段を具備することにある。

【0010】請求項2の発明の特徴は、前記スキャンに伴い、次に発生するピクセルを含む領域単位が前記描画対象領域内に入るか否かの領域判定を行う領域判定手段を具備し、少なくとも1個以上のピクセルが前記描画対象領域内に入る場合に、前記領域単位でピクセルを発生することにある。

【0011】請求項3の発明の前記領域判定手段は、直近で発生したピクセルを含む領域の外に設定する先行点が前記描画対象領域内に入るか否かで前記領域判定を行う。請求項4の発明の特徴は、前記領域内の少なくとも1個以上の特定のピクセルのデータと前記領域内の残りのピクセルのデータとを算出する際の計算精度を変化させる算出手段を具備することにある。

【0012】請求項5の発明の特徴は、前記領域内のピクセルデータを同時に処理する並列処理手段を具備することにある。

【0013】請求項6の発明の特徴は、前記並列領域を形成する各領域内の少なくとも1個以上の特定のピクセルのデータについては精度の高い計算により算出し、前記各領域内の残りのピクセルのデータについては精度の低い計算により算出することにある。

【0014】請求項7の発明の特徴は、前記領域単位で発生したピクセル各々が前記描画対象領域内に入っているかどうかを判定するマスク情報を発生するマスク発生手段と、前記発生したマスク情報により前記描画対象領域内に入っているピクセルだけを描画する描画手段を具備することにある。

【0015】請求項8の発明の特徴は、前記領域単位を任意の数、複数個連結して形成する並列領域を設定する設定手段と、前記描画対象領域内に前記設定された並列領域単位でピクセルデータを同時に発生する並列処理手段とを具備することにある。請求項9の発明の前記並列処理手段は、前記領域単位の少なくとも1個以上の特定のピクセルを代表点とし、且つこの代表点の位置を前記領域単位の連結個数によらずに一定とし、この代表点のデータと残りピクセルのデータを算出する際の計算精度を変化させる。

#### 【0016】

【発明の実施の形態】以下、本発明の実施の形態を図面に基づいて説明する。図1は本発明の画像処理装置の第1の実施の形態を示したブロック図である。画像処理装置10は、ホスト(CPU)1と装置を接続するインターフェース(I/F)3、DDA(Digital Differential Analyze)前処理を行うセットアップエンジン4、DDAに必要なDDA初期値、傾き等の情報を生成してオブジェクトのピクセルデータを生成するDDAブロック5、隠面消去等の処理を行った後、必要なデータをメモリ7に書き込むメモリアンターフェース6、ディスプレイ(CRT)9をコントロールするCRTコントローラ8から成っている。又、画像処理装置10のインターフェース(I/F)3がシステムバス2を介してCPU1に接続され、CRTコントローラ8がディスプレイ9に接続されている。

【0017】次に本実施の形態の動作について説明する。CPU1はジオメトリ処理や光源処理等を行ってオブジェクトの頂点情報を生成し、且つ描画に必要なデータを画像処理装置10に出力する。この際、頂点座標は小数部を何ビットか持った固定小数点フォーマットである。頂点情報はホスト側とのインターフェース(I/F)3を通じてDDA(Digital Differential Analyze)前処理を行うセットアップエンジン4に入力される。

【0018】セットアップエンジン4は頂点情報に基づき、DDAに必要なDDA初期値、傾き等の情報を生成してDDA処理を行うDDAブロック5に出力する。DDAブロックはオブジェクトのピクセルデータを生成し、メモリアンターフェース6に送る。メモリアンターフェース6では隠面消去等の処理を行った後、必要なデータをメモリ7に書き込む。メモリ7は1画面又は複数画面分のデータを保持するのに十分な容量を有し、各ピクセルのカラー情報の他、Zバッファリングのための奥行き値(Z値)なども必要に応じて格納することが可能である。

【0019】1画面分のデータ処理が完了した時点で、メモリ7のデータはメモリアンターフェース6を通じてCRTコントローラ8に送られ、CRTコントローラ8はsync等の同期信号と共にピクセルデータをディスプレイ9に出力して画像を表示する。

【0020】図2は図1に示したセットアップエンジンの詳細構成例を示したブロック図である。セットアップエンジン4は、頂点の座標情報からDDA描画開始頂点及びDDA描画方向(スキャン方向)を決定するブロック41、ブロック41で決定した描画開始頂点の座標(CPUから与えられる固定小数点フォーマット)からDDA描画開始点座標を算出するブロック42、先行する点の領域判定を行うためのDDA用のパラメータ(DDA初期値、DDA傾き)を算出するブロック43、カラー値等ピクセルデータ生成用のDDA初期値及びDDA傾きを算出するためのブロック44、これらブロック

を制御するコントローラであるブロック45から成っている。セットアップエンジン4は、バス50を通してインターフェース3からオブジェクトの頂点情報を受け取り、ブロック41はDDA描画開始点頂点及びDDA描画方向を決定する。

【0021】尚、101~104はそれぞれDDAブロック5への出力である。101はDDA描画方向、102はDDA描画開始点座標、103はデータDDA用パラメータ（初期値、傾き）、104は領域判定DDA用パラメータ（初期値、傾き）である。

【0022】但し、描画開始点座標は格子点座標であり整数値である。

【0023】図3を用いてDDAブロック5におけるスキヤンの制御方法について示す。描画対象のオブジェクトは図の開始辺201及び終了辺202と203で形成されるトライアングルである。ここで開始辺とはX軸に沿ったスキヤンがオブジェクトに侵入する側の辺をいい、終了辺とはスキヤンが抜けて行く側の2辺のことをいう。

【0024】X軸に沿ったDDAの方向は図3の31に示されるように正方向、またY軸に沿ったDDAの方向は図3の32に示されるように負の方向である。図3の各矩形領域は同時に処理されるピクセルのまとまり（スタンプと称する）を表し、特にスタンプ33はDDA開始スタンプである。

【0025】図3中のスタンプ33内の各丸印はピクセルを表し、4×2のピクセル群から構成されている。その内、ピクセル3Pはセットアップエンジン4で決定されるDDA描画開始点座標のピクセルである。また、この装置では図示されていないシザリング領域を設定することが可能である。

【0026】図3のスキヤンはDDA開始スタンプ33からXの正方向に進み、あるところでそのラインのスキヤンを終了し、次のラインに移る（Yの負方向に進む）。これを繰り返してオブジェクトの全ての領域をスキヤンし終わったら、そのオブジェクトの処理を終了する。

【0027】尚、X方向のスキヤンは（1）X終了側のシザリング領域から抜ける。（2）Y開始側のシザリング領域に達していない。（3）図3の終了辺（202、203）から抜ける条件のどれかが成立した場合に終了する。

【0028】また、次ラインのスキヤン開始位置は、（1）現在のスキヤンラインの開始位置、（2）現在のラインの開始辺への侵入座標、（3）シザリング領域のXの開始側エッジへの侵入座標の内、Xの描画方向に向かって最も奥側の座標が選択される。

【0029】上記のような制御を行う上で開始辺201、終了辺202、203に対する領域判定が必要になるが、現在処理しているスタンプに対して図4に示すよ

うな先行したピクセル（先行点）に対して領域判定が行われる。

【0030】図4において、201は開始辺、202、203はそれぞれ終了辺である。描画方向はX軸に沿って正方向（31）、Y軸に沿って負方向（32）である。又、ピクセル4P、5P、6Pが領域判定を行う先行点で、4Pは開始辺201に対する領域判定に用いる先行点、5Pと6Pは終了辺202、203に対する領域判定に用いる先行点で、2つの終了辺に対してより遠い側が判定ポイントとして選択される。図4では、辺202に対して6Pが、辺203に対して5Pが選択される。

【0031】上記した開始辺201、終了辺202、203に対する領域判定は先行点の各辺に対する直線方程式を解くことによって行う。点（x0, y0）を通り、x, yの傾きがそれぞれd1dx, d1dyである辺の直線方程式は $l = d1dx \times (x - x0) + d1dy \times (y - y0)$  (1)

のように書ける。ここで、DDA開始スタンプに先行させた点の領域判定対象である辺に対する直線方程式の解をl1, l2, l3とする。

【0032】l1, l2, l3を基点として描画スタンプが移動する毎に4×d1dx, 或いは2×d1dyを加減算するDDAを行うことにより、各矩形領域（スタンプ）の先行させる点の直線方程式の解を算出し、解の符号から先行点の各辺に対する領域判定を行う。

【0033】図4の先行点4Pを例に挙げると、辺201に対して次のように領域判定ができる。

【0034】if (d1dx > 0) の時、

1) 解の符号が正なら先行点は辺201を超えた。

2) 解の符号が負なら先行点は辺201まで達していない。

else (d1dx < 0) の時、

3) 解の符号が正なら先行点は辺201まで達していない。

4) 解の符号が負なら先行点は辺201を超えた。

【0035】辺に対する領域判定を現スタンプ内のピクセルでなく先行させた点で行うことで、次のような利点がある。開始辺に対しては図5に示すように開始辺の傾き角がX軸に対して非常に緩やかになった時に生じるような次ラインのスキヤンの無駄を低減することができる。

【0036】図5のスタンプ51に着目し、開始辺に対する領域判定を現スタンプ内のピクセル8Pで行うと、次ラインのスキヤンはスタンプ52から開始され、実際にはスタンプ52内のピクセルは全て領域外なので、スタンプ52は無駄になる。しかし、スタンプ51の先行点4Pで開始辺に対する領域判定を行った場合、次ラインはスタンプ53からスキヤンを開始できるので、無駄なスタンプの発生が無くなる。また、終了辺に対しては

そのラインのオーバースキャンを低減できる。これらにより、1つのオブジェクトに要する描画のための演算サイクルを減らすことができ、より高速な描画が可能になる。

【0037】次にセットアップエンジン4におけるデータDDAの傾き、及びDDA描画開始点のデータ算出方\*

$$P = dpdx \times x + dpdy \times y + c \quad (dpdx, dpdy, c \text{ は定数})$$

(3)

オブジェクトとしてトライアングルを処理する場合、3 頂点  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$  ※に対してピクセルデータが  $p_0, p_1, p_2$  であつたとすると、

$$dpdx = \{ (p_1 - p_0) \times (y_2 - y_0) - (p_2 - p_0) \times (y_1 - y_0) \} / e_0 \quad (4)$$

$$dpdy = \{ (p_2 - p_0) \times (x_1 - x_0) - (p_1 - p_0) \times (x_2 - x_0) \} / e_0 \quad (5)$$

但し、 $e_0 = (x_1 - x_0) \times (y_2 - y_0) - (x_2 - x_0) \times (y_1 - y_0)$  である。

【0039】 $dpdx$  は  $x$  方向のデータの傾き、 $dpdy$  ★

$$ps = dpdx \times (xs - xs_0) + dpdy \times (ys - ys_0) + ps_0$$

(6) である。

【0040】このシステムは  $4 \times 2$  ピクセルのデータの並列演算を行うが、DDAブロックは図3のピクセル5であるDDA描画開始点のデータ  $ps$  から同一スタンプ内の他のピクセルデータを生成する。この場合、次のような演算が行われる。

【0041】

$$ps - dpdx$$

$$ps + dpdx$$

$$ps + 2 \times dpdx$$

$$ps - dpdy$$

$$ps - dpdy - dpdx$$

$$ps - dpdy + dpdx$$

$$ps - dpdy + 2 \times dpdx \quad (7)$$

式(6)、(7)で求めた描画開始スタンプの全ピクセルのデータ及び式(4)、(5)で求めたDDAの傾きとから、各描画スタンプのピクセルを並列に生成する。

【0042】図6はこれを実現する回路例であり、この回路は領域判定DDAにも適応できる回路である。1種類のデータを8並列で同時処理するために図6のようなユニットを8個持つ。

【0043】レジスタ61と62は式(4)と式(5)の傾き  $dpdx$ ,  $dpdy$  をそれぞれ4倍、2倍した値を保持する。セレクト64はセットアップエンジン4により算出されるデータである式(6)のデータ603、又はDDAブロックで算出される式(7)のデータを入力する。

【0044】セレクト64はデータ603か、或いは前のピクセルのデータである611を選択するが、描画開始時のみデータ603が選択され、それ以外はデータ611が選択される。

【0045】レジスタ66はセレクト64の出力を毎演

\*法について示す。オブジェクトのピクセルデータは先の頂点情報を線形補間することによって求めることができる。このような時、ピクセルデータ  $P$  はスクリーン座標  $(x, y)$  に対して次のような関係になる。

【0038】

★  $y$  は  $y$  方向のデータの傾きである。DDA描画開始点  $(xs, ys)$  のデータ  $ps$  は、3頂点のうち描画開始

頂点の座標  $(xs_0, ys_0)$  及びデータ  $ps_0$  から

算サイクル更新して格納するレジスタであり、この値612はDDAの出力である。レジスタ65の内容は毎スキャンラインの始めにセレクト64の出力が書き込まれて更新されるが、それ以降は先行点がシザリングのX開始側エッジか、若しくは開始辺を超えたと判定した位置でのセレクト64の出力を書き込むのみである。フラグ609はスキャンがX方向に進むか、またはY方向に進むかを表すフラグで、セレクト67、68に共通である。セレクト67と68の選択動作は以下の如くである。

30 【0046】セレクト67はスキャンがX方向に進む時、レジスタ61の出力を選択し、スキャンがY方向に進む時、レジスタ62の出力を選択する。セレクト68はスキャンがX方向に進む時、レジスタ66の出力を選択し、スキャンがY方向に進む時、レジスタ65の出力を選択する。

【0047】加減算器70はセレクト67と68の出力を加減算するが、これによりスキャンがX方向に進む時、直前のスタンプの解に対して  $4 \times dpdx$  を加算(減算)することによって、新たなピクセルデータを生成し、次のラインに移る時はレジスタ65に保持しておいた値に対して  $2 \times dpdy$  の加算(減算)することによって、そのラインのスタート位置の初期値を得る。

【0048】図7にデータ生成の模式図を示す。201は描画オブジェクトの開始辺、203は終了辺の1つである。各円はピクセルを表し、その中心がピクセル中心である。描画方向はX軸に沿って正方向、またY軸に沿っては負方向である。この描画オブジェクトは各矩形でまとめられている  $4 \times 2$  ピクセルの矩形領域の単位(スタンプ)でデータ処理される。3P~7Pはそれらの矩形領域に含まれるピクセルの1つであり、複数ある図6

で示した演算ユニットのうち同じユニットで生成されるピクセル群である。

【0049】ピクセル3Pを含む矩形領域が描画開始スタンプになる。ピクセル3Pのデータを基に、ピクセル4Pの格子点のピクセルデータが生成される。この2ラインのスキューはこのスタンプで終了し、次の2ラインへ移る。

【0050】ピクセル3Pを含むスタンプはその先行点8Pが既に開始辺201を超えているので、次ラインのスキューはピクセル5を含むスタンプから開始される。\*10

$$\begin{aligned} \text{ピクセル4Pのデータ } p_4 &= p_3 + 4 \times d_{pdx} \\ \text{ピクセル5Pのデータ } p_5 &= p_3 - 2 \times d_{pdy} \\ \text{ピクセル6Pのデータ } p_6 &= p_3 - 2 \times d_{pdy} + 4 \times d_{pdx} \\ \text{ピクセル7Pのデータ } p_7 &= p_3 - 2 \times d_{pdy} + 4 \times d_{pdx} - 2 \times d_{pdy} \end{aligned} \quad (8)$$

この(8)式の4倍や2倍の演算はシフト動作で実現できることから、(8)式は実質、加減算器のみで構成できる。

【0052】以上のようにして格子点座標のデータのみの演算を行ってピクセルデータを発生していく。これは従来のような辺方向のデータの傾きからオブジェクトの辺上のデータ(X軸に平行なグリッドと辺の交点のデータ)をまず求め、続いてX方向に線形補間してスキューライン上のデータの生成を行うような画像処理装置に比べて、演算リソース節約の点で有利である。

【0053】従来のように、毎スキューライン毎に辺上のデータを求めたとすると、X軸に平行なグリッドと辺の交点は必ずしも格子点上にないため、X方向のスキューの前に上記の辺上のデータから近傍の格子点のデータを求める処理が必要になる。これにはサブピクセル分の補正を行うための乗算器が必要である。更に、各ピクセル※

$$\begin{aligned} l_{2s} &= l_{11} + 2 \times d_{l1dy} - 3 \times d_{l1dx} \\ l_{3s} &= l_{11} + 2 \times d_{l1dy} - 2 \times d_{l1dx} \\ l_{4s} &= l_{11} + 2 \times d_{l1dy} - d_{l1dx} \\ l_{5s} &= l_{11} + 2 \times d_{l1dy} \\ l_{6s} &= l_{11} + d_{l1dy} - 3 \times d_{l1dx} \\ l_{7s} &= l_{11} + d_{l1dy} - 2 \times d_{l1dx} \\ l_{8s} &= l_{11} + d_{l1dy} - d_{l1dx} \\ l_{9s} &= l_{11} + d_{l1dy} \end{aligned} \quad (9)$$

ここで、 $d_{l1dx}$ 、 $d_{l1dy}$ はそれぞれ開始辺の直線方程式のX、Yの傾きである。

【0056】同様に図8のピクセル10P、即ち終了辺に対する領域判定用のピクセルの方程式の解を $l_{10}$ と★

$$\begin{aligned} l_{2e} &= l_{10} + d_{l2dy} - 4 \times d_{l2dx} \\ l_{3e} &= l_{10} + d_{l2dy} - 3 \times d_{l2dx} \\ l_{4e} &= l_{10} + d_{l2dy} - 2 \times d_{l2dx} \\ l_{5e} &= l_{10} + d_{l2dy} - 1 \times d_{l2dx} \\ l_{6e} &= l_{10} - 4 \times d_{l2dx} \\ l_{7e} &= l_{10} - 3 \times d_{l2dx} \\ l_{8e} &= l_{10} - 2 \times d_{l2dx} \end{aligned}$$

\*ピクセル5のデータはピクセル3Pのデータから生成され、ピクセル6Pのデータはピクセル5Pのデータから生成される。この2ラインのスキューが終了すると、再び次のラインにスキューは移動し、ピクセル7Pのデータがピクセル6Pのデータから生成される。ピクセル3Pのデータを $p_3$ とすると、各ピクセル4P～7Pのデータ $p_4 \sim p_7$ はデータの傾き $d_{pdx}$ 、 $d_{pdy}$ を使って、次のように表される。

【0051】

※ルのデータの種類のY方向の並列度が増すと、それだけこの乗算器が必要となり、多くの演算リソースを要する。しかるに、本実施の形態の装置ではこのような乗算器は必要ない。従って演算リソースを節約した小型の画像処理装置を実現することが可能である。

【0054】図8はピクセルが描画領域内に存在するかどうかを表すマスクの生成について示したものである。矩形単位で並列処理した時に描画領域からはみ出したピクセルのデータも生成するので、このデータを後段で除外するためにマスクが必要である。各丸形はピクセルを示し、2P～9Pは同時生成されるピクセル群を示し、1Pは開始辺に対する領域判定を行うピクセルである。ピクセル1Pの開始辺に対する直線方程式の解を $l_{11}$ とする。 $l_{11}$ を基に、ピクセル2P～9Pの直線方程式の解 $l_{2s} \sim l_{9s}$ を算出することができる。

【0055】即ち、

★すると、ピクセル2Pからピクセル9Pの終了辺に対する直線方程式の解 $l_{2e} \sim l_{9e}$ を算出することができる。

【0057】即ち、

$$l9e = l10 - 1 \times d l2dx$$

ここで  $dl2dx$ 、 $dl2dy$  はそれぞれ終了辺の直線方程式の X と Y の傾きである。式 (9) と式 (10) の解の符号からピクセル 2P ~ 9P のそれぞれの辺に対する領域判定ができる。

【0058】その結果、各ピクセルにおいて、式 (9) と式 (10) の結果に基づき、両辺に対してそのピクセルが描画領域側にあると判定されたら描画対象としてマスク生成、そうでなければ描画非対象としてマスク生成となる。メモリインターフェース 6 はピクセルデータとマスク情報を受け取り、マスクから描画対象と判断されるピクセルに対してオペレーションを実行する。

【0059】本実施の形態によれば、X 方向、Y 方向にスキャンして X、Y 方向の傾きを用いて格子点のデータのみを発生するため、加減算器のみでポリゴン内のピクセルを発生することができ、ハードウェアリソース (DDA ブロック 5) を低減することができる。

【0060】又、スタンプ単位で、X、Y 方向にスキャンしながらピクセルを発生する際に、ピクセルが発生領域内にあるかどうかの判定を現在処理中のスタンプ内のピクセルを用いるのではなく、このスタンプの外の先行したピクセル (先行点) を用いることにより、無駄なスキャンを無くことができ、効率的にピクセルを発生することができる。

【0061】又、発生したピクセルは矩形領域 (スタンプ) 単位であるため、ピクセル発生領域からはみ出してしまうピクセルが発生するが、発生したピクセルがピクセル発生領域にあるかどうかをマスクにより判定し、ピクセル発生領域内のピクセルのみを描画する。

【0062】尚、上記したスキャンの方法は描画速度を速めるためにスタンプ単位で行ったが、最低 1 個のピクセル単位で行うこともでき、この場合は、ピクセル発生領域からはみ出してしまうピクセルがなくなるため、上記したマスクは必要なくなる。

【0063】図 9 は本発明の画像処理装置の第 2 の実施の形態を示したブロック図である。演算ユニット 81 はスタンプ内の代表点のデータを算出するもので、図 6 に示した回路と同等のものである。この演算ユニット 81 にはセットアップエンジンで生成される X 方向のデータの傾きと Y 方向のデータの傾き 101、102 がそれぞれ 4 倍、2 倍されて入力される。ユニット 82 はデータの傾き 101、102 の小数部下位ビットを切り捨て、精度を落とした傾きを生成するためのユニットであり、このユニット 82 の出力 801 と 802 は精度を落とした X、Y のそれぞれの傾きである。ユニット 83 はユニット 81 の出力 803 (803 は代表点のデータである) と精度を落とした X、Y の傾き 801、802 から代表点以外の残りのピクセルデータを生成するユニットである。このユニット 83 の出力データである 804 及び 803 が 1 並列分のデータになる。

(10)

【0064】ここで、101 と 102 はセットアップエンジンから送られてくるデータの X 方向と Y 方向の傾きであり、下位 10 bit が小数部である固定小数点フォーマットになっている。103 は DDA 開始点のデータで、セットアップエンジンにて生成される。この DDA 開始点の座標は格子点上の座標である。演算ユニット 81 には X 方向のデータの傾きと Y 方向のデータの傾きがそれぞれ 4 倍、2 倍されて入力される。

【0065】但し、801 と 802 は小数部 4 ビットの固定小数点フォーマットである。小数部のビット数は通常、傾きの精度を十分取った時の値に比べて ±1 (整数値) 以上の誤差が出ないように決められる。尚、本実施の形態は DDA ブロックについてののみ示してあるが、他の構成は図 1 に示した第 1 の実施の形態と同様である。

【0066】次に本実施の形態の動作について説明する。この DDA ブロックを含む画像処理装置は 1024 × 1024 ピクセルの座標系をサポートする。また、4 × 2 ピクセル単位での並列処理が可能である。更にこの DDA ブロックで上記並列処理するピクセル群 (スタンプ) は DDA 開始点のデータ及び精度の高い傾きから生成される点 (代表点) を 1 点含み、残りの 7 点は代表点と精度の低い傾きから生成される。ここでいう精度の低い傾きとはセットアップエンジンで生成されたデータの傾きの小数部下位ビットを切り捨てたものである。

【0067】図 10 は描画オブジェクトに対するスタンプを表したものである。このオブジェクトはトライアングルであるが、201 と 203 はそれぞれその開始辺と終了辺の一部分である。31 と 32 はそれぞれ X、Y の描画方向である。各円はその中心がピクセル中心であるひとつのピクセルを表し、5P は描画開始点であり、そのデータはセットアップエンジンで生成される。6P は 5P からデータが生成されるスタンプ代表点である。ピクセル 6P のデータ p6 はピクセル 5P のデータ p5 と次のような関係にある。

【0068】

$$p6 = p5 + 4 \times dpdx \quad (11)$$

但し、 $dpdx$  はセットアップエンジンで生成される X 軸に沿ったデータの傾きである。このほかの網目状に塗りつぶされたピクセルも代表点に相当する。これらの代表点は図 9 のユニット 81 で、データ p5 及び  $dpdx$ 、 $dpdy$  から生成される。ここで  $dpdy$  はセットアップエンジンで生成される Y 軸に沿ったデータの傾きである。

【0069】図 9 のユニット 82 における傾きの小数部切り捨て方法について説明する。セットアップエンジンで生成されるデータの傾きを  $dpdx$ 、 $dpdy$  とし、それぞれの精度を落とした傾きを  $dpspx$ 、 $dpsdy$  とする。

【0070】もし ( $dpdx \geq 0$ ) なら



$dpsdx = (dpdx \text{ の下位 6 ビットを切り捨て})$

もし  $(dpdx < 0)$  なら

$dpsdx = (dpdx \text{ の下位 6 ビットを切り上げ})$

もし  $(dpdy \geq 0)$  なら

$dpsdy = (dpdy \text{ の下位 6 ビットを切り捨て})$

もし  $(dpdy < 0)$  なら

$dpsdy = (dpdy \text{ の下位 6 ビットを切り上げ})$

このようにデータの増減の方向によって小数部下位ビットを切り捨てる際の丸め込みの方法を決めることによ

り、この低い精度の傾きからデータを生成した時に、 $\pi \times 10$

$$p1 = pd4 - dpsdy - dpsdx$$

$$p2 = pd4 - dpsdx$$

$$p3 = pd4 - dpsdy$$

$$p5 = pd4 - dpsdy + dpsdx$$

$$p6 = pd4 + dpsdx$$

$$p7 = pd4 - dpsdy + 2 \times dpsdx$$

$$p8 = pd4 + 2 \times dpsdx$$

但し、上記(12)式中の2倍演算はシフト動作と等価なので実質加減算器のみで構成される。

【0073】ピクセルデータを発生させるために、それぞれ1つの精度の傾きで全演算を行おうとすると、座標空間の広さを考えた高い精度の演算が要求されるが、本実施の形態によれば、代表点近傍のピクセル生成に限定した低い精度の傾きを用意し、スタンプ内の代表点以外のデータは代表点のデータ及びこの低い精度の傾きから生成するため、演算リソースを節約することが可能になり、その分、ハードウェアの規模を小さくでき、画像処理装置を小型化することができる。

【0074】図12は本発明の画像処理装置の第3の実施の形態を示したブロック図である。画像処理装置のDDAブロックは、セットアップエンジンから入力される傾き301を定倍した傾きを生成するブロック121、セットアップエンジンから入力されるDDAの初期値302とブロック121から出力される精度の高い傾きから複数の小矩形領域の主代表点(1点)のデータを生成するブロック122、主代表点のデータとブロック121が出力する精度の高い傾きから前記複数の小矩形領域の残りの代表点を生成するブロック123、精度の高い傾き301に対して、小数部の下位ビットを切り捨てて精度を落としたデータの傾きを生成するブロック124、ブロック122で生成される主代表点のデータ309及びブロック124で生成された精度の低いデータの傾き307とから前記主代表点が含まれる小矩形領域の主代表点以外のデータを生成するブロック125、ブロック123で生成される代表点のデータ311及びブロック124の精度の低いデータの傾き307とからその代表点が含まれる小矩形領域の代表点以外のデータを生成するブロック126から成っている。

【0075】尚、本例は矩形領域を構成する複数のピクセル群を単位として並列度を変更しても各ピクセルの演

\* クセルの連続する方向へのデータの単調増加、単調減少が保たれる。

【0071】図11は図9のブロック83で行うスタンプ内の代表点以外のデータを算出する方法を示した図である。代表点4Pのデータをpd4とすると、それ以外のピクセル1P~3P、5P~8Pのデータp1~p3、p5~p8は前記のdpsdx、dpsdyを用いて次のように求めることができる。

【0072】

(12)

算結果が変わらないDDA処理例であり、他の構成は図1に示した画像処理装置と同様である。

【0076】次に本実施の形態の動作について説明する。本例のDDA処理を含む画像処理装置は1024×1024ピクセルの座標系をサポートする。また、このDDAブロックの機能は次のような幅を持った矩形領域に相当するピクセルのデータを並列処理でき、主代表点(精度の高い傾きからデータが算出される点)として次のような個数を持つ。

【0077】(1) X方向に4ピクセル、Y方向に2ピクセルの幅、主代表点は1個

(2) X方向に8ピクセル、Y方向に2ピクセルの幅、主代表点は2個

(3) X方向に4ピクセル、Y方向に4ピクセルの幅、主代表点は2個

(4) X方向に8ピクセル、Y方向に4ピクセルの幅、主代表点は4個

それぞれの矩形領域は4×2のピクセルから成る小矩形領域を1~4個含み、それぞれの小矩形領域は代表点をひとつずつ含む。

【0078】301はセットアップエンジンにて算出されるデータのX、Y方向の傾きdpdx、dpdyであり、小数部10ビットの固定小数点フォーマットである。302は同じくセットアップエンジンで算出されるDDAの初期値であり、並列度に拘らず1点分算出される。

【0079】ブロック121は入力される傾き301を定倍してブロック122及びブロック123に出力する。この場合、ブロック122への出力に対しては並列度に応じて倍数が変わる。また、ブロック123への出力はdpdx、dpdyをそれぞれ4倍、2倍にして出力する。

【0080】図13はブロック121からブロック12

は図12のブロック125においてピクセル3Pのデータ及び精度の低い傾き307から生成される。このために式(12)と同様な演算を行うための演算ユニットが図12のブロック125に含まれている。矩形領域9〜11の代表点以外のデータはブロック126にてそれぞれ代表点4P〜6Pのデータ及び精度の低い傾きから式(12)と同様な演算に基づき生成される。

【0089】これらの演算を行うユニットが図12のブロック126に含まれている。最大で上記の(4)に相当するピクセルを同時処理する必要があるのでブロック126はブロック125の3倍の演算器を含んでいる。

【0090】矩形領域分のデータ307が生成されると、Xの正方向にスキャンは進み、次の主代表点ピクセル12Pのデータがピクセル3Pのデータから図12のブロック122にて生成され、以下同じように8×4ピクセル分のデータが並列処理される。

【0091】X方向のスキャンが終了すると次のラインに移り、主代表点であるピクセル13Pのデータがピクセル3Pのデータと精度の高い傾きから生成され、以下同じように矩形領域内の8×4ピクセルのデータが生成される。

【0092】図16は上記の並列度が(2) (8×2ピクセル並列演算)の場合におけるピクセルデータ生成の様子を示している。太線の矩形領域161が並列処理する8×2のピクセルに相当する。この矩形領域は4×2ピクセルからなる小矩形領域を2つ含み、ピクセル2Pとピクセル3Pはそれぞれの代表点(特にピクセル2Pは主代表点)である。

【0093】ここで、各小矩形領域の境界は図15のそれと同じであり、また各代表点の位置も同じである（図15のピクセル3Pと図16のピクセル2Pの座標は同じである）。

【0094】図16にて、それぞれの代表点と精度の低い傾きから矩形領域の残りのデータが生成されると、スキャンは次に移り、次の主代表点ピクセル4 Pのデータがピクセル2 Pから生成される。以下、同じように処理が進み、このラインの処理が終わると、スキャンは次の2ラインに移る。主代表点5 Pのデータがピクセル2 Pのデータから生成され、同じようにこのラインの処理が開始される。

【0095】本実施の形態によれば、 $4 \times 2$ ピクセルの小矩形領域を単位として演算の並列度を変えることができるが、その際に矩形領域に含まれる各小矩形領域の境界とそこに含まれる代表点の位置を並列度によらず同じにし、これら代表点のデータを求める際の精度と、それ以外の点のデータを求める際の精度を並列度に依らず、それぞれ同じになるようにすることができる。これにより並列度を変えられる本例の装置において、並列度が変わっても各ピクセルの演算結果は同じになることを保証することができる。

【0088】小矩形領域8のピクセル3 P以外のデータ 50

【0096】尚、上記実施の形態では、小矩形領域は4×2ピクセルであり、この小矩形領域が少なくとも1個以上集まって矩形領域を形成するが、前記小矩形領域は必ずしも、4×2ピクセルである必要はない。

【0097】

【発明の効果】以上詳細に説明したように、請求項1、2の発明によれば、少なくとも1個以上のピクセルを含む領域単位でスクリーン座標系のX軸、Y軸に沿って描画対象領域をスキャンすることにより、ピクセルを発生するためのデータ演算で格子点以外の座標のデータを扱わず格子点上のデータのみを加減算器のみで算出することができ、ハードウェアリソースを大幅に低減することができる。

【0098】請求項3の発明によれば、現在処理している領域の外に位置する先行点により、前記描画対象領域の外であるか内であるかの領域判定を行うため、現スキャン方向のオーバースキャンを防止することや、次ラインの無駄なスキャンの低減が可能になり、描画の高速な画像装置を実現できる。

【0099】請求項4の発明によれば、領域内の特定のピクセルのデータを精度の高い計算により算出し、前記領域内の残りのピクセルのデータを低い精度の計算により算出することにより、ハードウェアリソース低減すると共に、ピクセルの描画速度を向上させることができる。

【0100】請求項5の発明によれば、領域内のピクセルのデータを同時に処理して発生させることにより、ピクセルの描画速度を向上させることができる。

【0101】請求項6の発明によれば、複数の領域内の特定のピクセルのデータを精度の高い計算により算出し、前記領域内の残りのピクセルのデータを低い精度の計算により算出することにより、ピクセルの描画速度を向上させることができると共に、ハードウェアリソースの増加を抑制することができる。

【0102】請求項7の発明によれば、マスク情報により、前記描画対象領域からはみ出したピクセルを取り除いて、前記描画対象領域内のピクセルのみを描画することにより、シャープな輪郭の画像を得ることができる。

【0103】請求項8の発明によれば、領域単位の連結個数を変更して、並列領域の並列度を変更することができる。

【0104】請求項9の発明によれば、領域単位を連結して形成した並列領域の代表点の位置を前記領域単位の連結個数に拘らず一定とすることにより、並列度を変更しても各ピクセルデータの算出結果を同じにすることができる。

【図面の簡単な説明】

【図1】本発明の画像処理装置の第1の実施の形態を示したブロック図である。

【図2】図1に示したセットアップエンジンの詳細構成例を示したブロック図である。

【図3】図1に示したDDAブロックによるピクセル発生動作例を示した説明図である。

【図4】図1に示したDDAブロックによる先行点を用いた領域判定動作を説明する説明図である。

【図5】図1に示したDDAブロックによる先行点を用いた領域判定のメリットを説明する図である。

【図6】オブジェクト内のピクセルを並列的に発生させるための回路例を示したブロック図である。

【図7】図6に示した回路によりピクセルを並列的に同時発生する動作を説明する説明図である。

【図8】ピクセルが描画領域内に存在するかどうかを表すマスクの生成動作を説明する図である。

【図9】本発明の画像処理装置の第2の実施の形態を示したブロック図である。

【図10】図9に示したDDA装置によるピクセルの発生動作を説明する説明図である。

【図11】図9の装置によりスタンプ内の代表点以外のデータを算出する方法を説明する説明図である。

【図12】本発明の画像処理装置の第3の実施の形態を示したブロック図である。

【図13】図12に示したブロック12への入力データを示した表図である。

【図14】図12に示したブロック13で算出される代表点のデータ例を示した表図である。

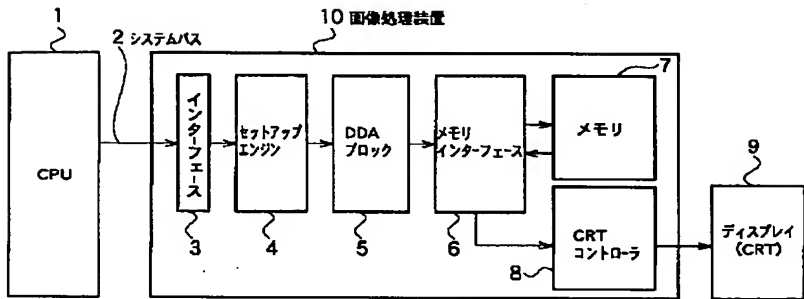
【図15】図12に示した装置に代表点を発生させる動作を説明する図である。

【図16】8×2ピクセル並列度の場合のピクセルデータの生成動作を説明する説明図である。

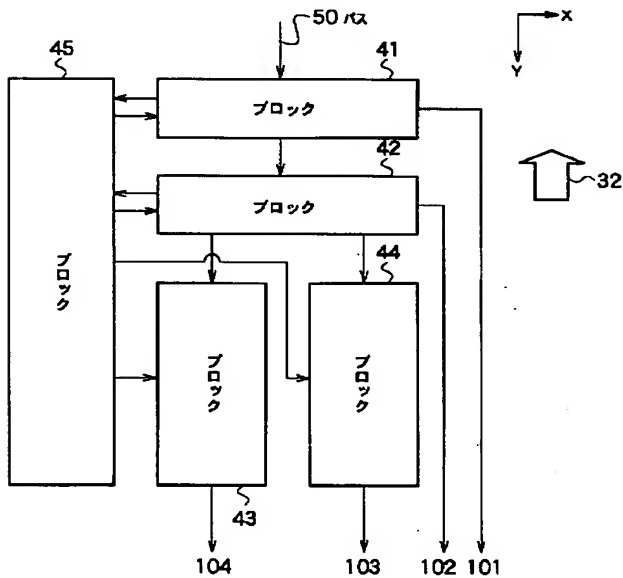
【符号の説明】

- 1 CPU
- 2 システムバス
- 3 インターフェース
- 4 セットアップエンジン
- 5 DDAブロック
- 6 メモリインターフェース
- 7 メモリ
- 8 CRTコントローラ
- 9 ディスプレイ (CRT)
- 10 画像処理装置
- 33、51、52、53 スタンプ
- 41～45、121～126 ブロック
- 50 バス
- 61、62、65、66 レジスタ
- 64、67、68 セレクタ
- 70 加減算器
- 81、82、83 ユニット

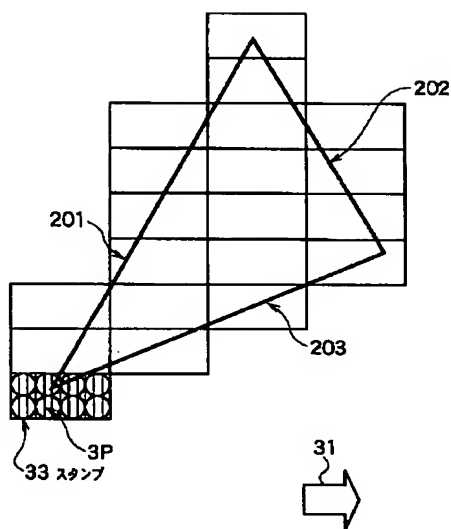
【图 1】



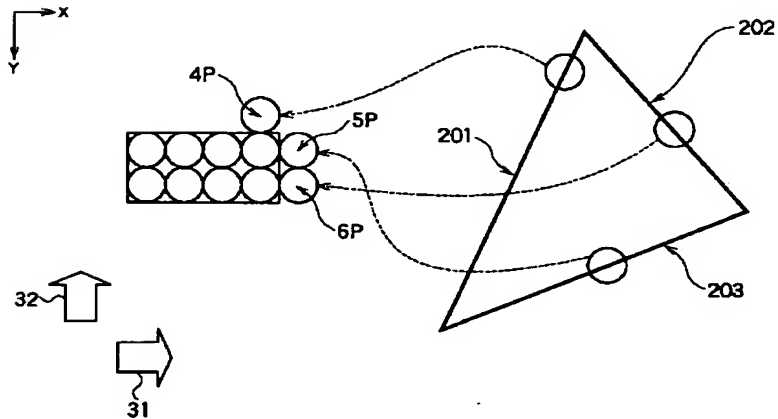
【图 2】



【图 3】



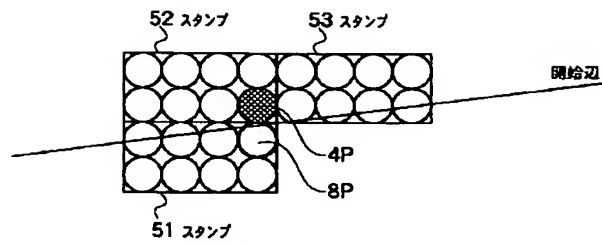
【図 4】



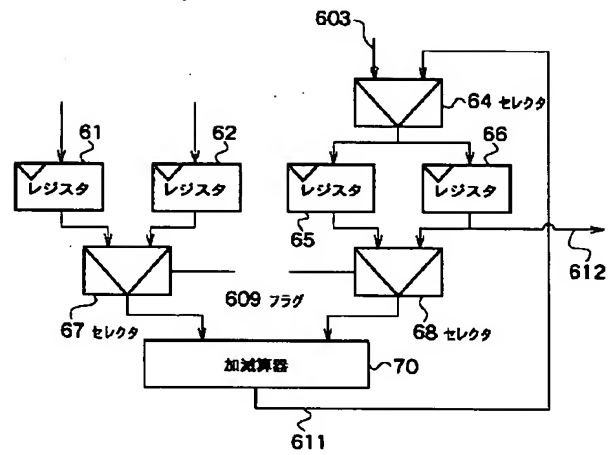
【図 13】

並列度	X方向の傾き	Y方向の傾き
(1)	$\text{dpdx}$ の4倍	$\text{dpcy}$ の2倍
(2)	$\text{dpdx}$ の8倍	$\text{dpcy}$ の2倍
(3)	$\text{dpdx}$ の4倍	$\text{dpcy}$ の4倍
(4)	$\text{dpdx}$ の8倍	$\text{dpcy}$ の4倍

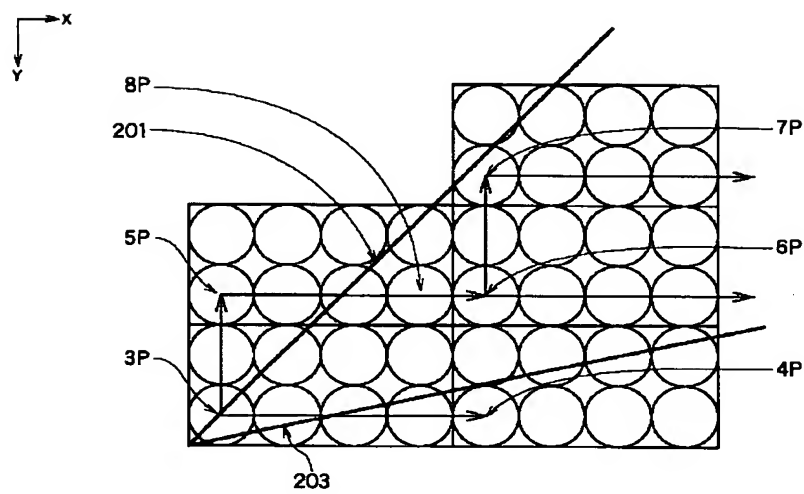
【図 5】



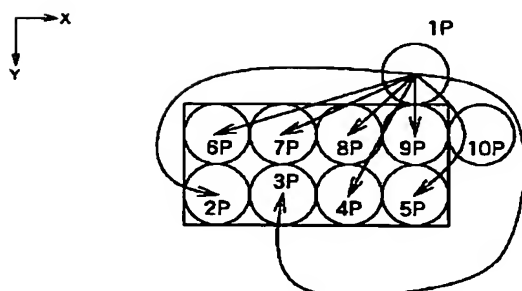
【図 6】



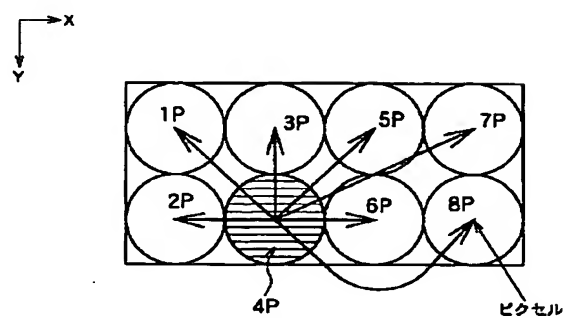
【図 7】



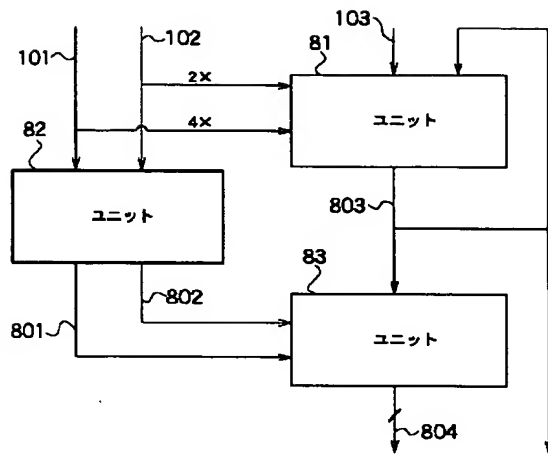
【図 8】



【図 11】



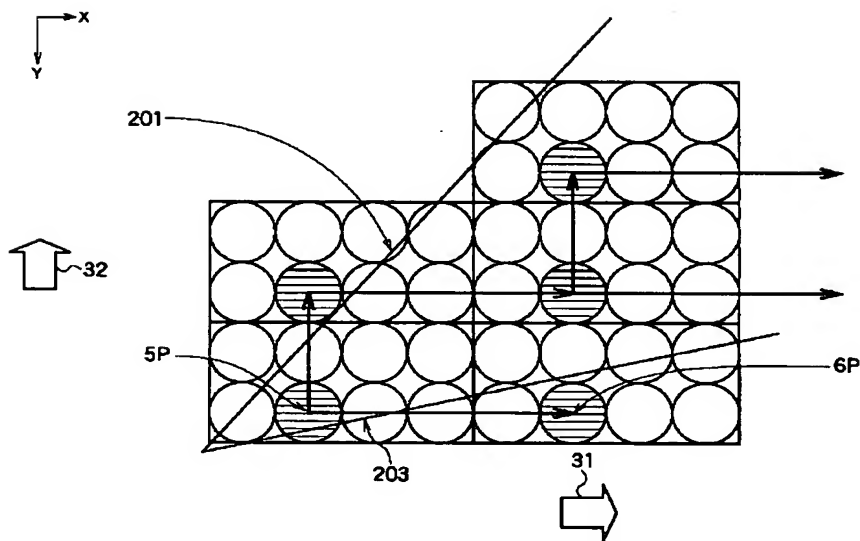
【図 9】



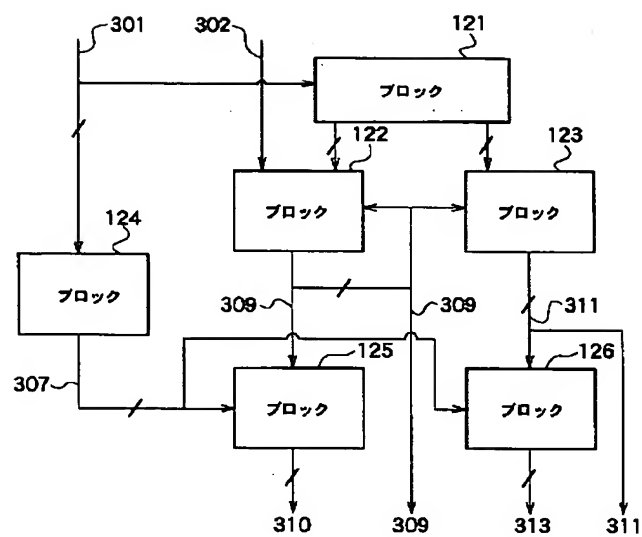
【図 14】

	並列度
$p4 = pd3 + 4 \times dpdx$	(2)×(4)の順に演算
$p6 = pd3 - 2 \times dpdy$	(3)×(4)の順に演算
$p8 = pd3 + 4 \times dpdx - 2 \times dpdy$	(4)の順に演算

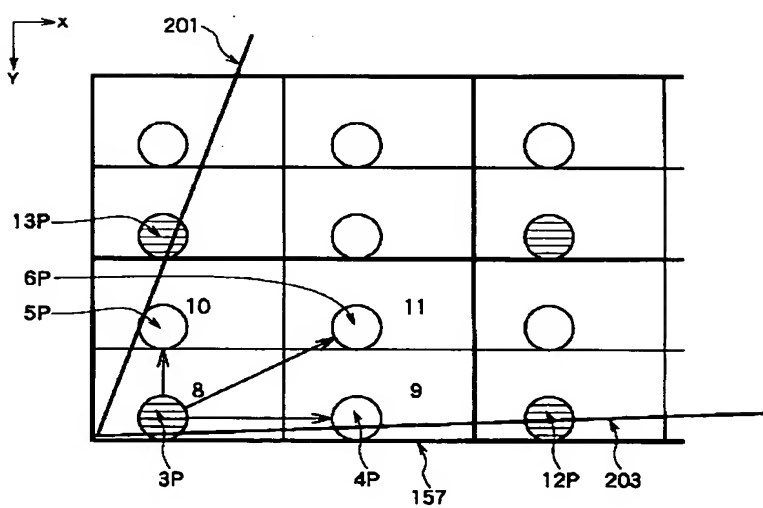
【図 10】



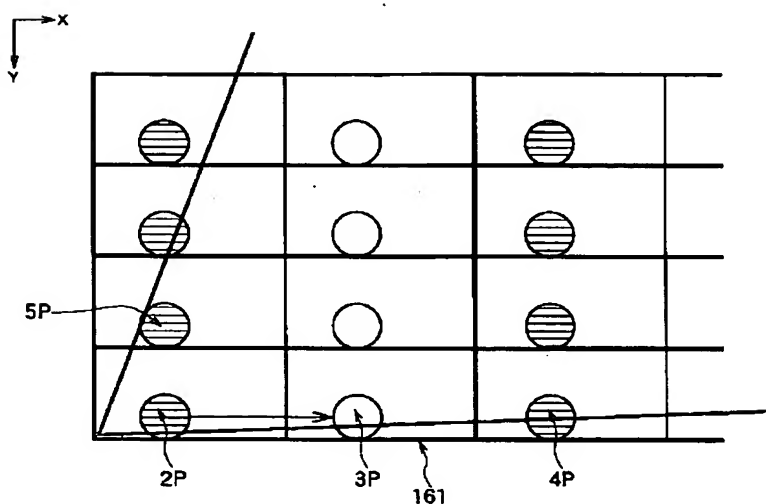
【図 12】



【図 15】



【図 16】



---

フロントページの続き

(72)発明者 竹中 康晴  
神奈川県川崎市幸区堀川町580番1号 株  
式会社東芝半導体システム技術センター内

Fターム(参考) 5B057 CA01 CA13 CB01 CB12 CD06  
CD14 CH04  
5B080 AA13 BA04 BA05 CA03 FA16  
5C082 AA01 BA12 CA18 CB01 DA22  
MM02 MM04